CSC3320 System Level Programming

Lab Assignment 10 - Post-Lab

Due at 11:59 pm on Friday, April 02, 2021

Purpose: Learn how to use the pointers to represent strings in C.

Part 1:

Write a function about string copy, the *strcpy* prototype "*char* strcpy (char* strDest, const char* strSrc*);". Here *strDest* is destination string, *strSrc* is source string.

1) Write the function *strcpy*, don't call C string library.

```
char* strcpy (char* strDest, const  char* strSrc){
        for(int i = 0 ;i != '\0'; i++){
                strDest[i] = strSrc[i];
                strDest[i] = '\0';
                return strDest;
        }
}
```

2) Here *strcpy* can copy *strSrc* to *strDest*, but why we use *char** as the return value of *strcpy*?

Since we need to return character array, named strDest, char* must be used to return character array.

Part 2:

Write a program ***findStr.c*** that finds the "smallest" and "largest" in a series of words. **After the user enters the words, the program will determine which words would come first and last if the words were listed in dictionary order**. The program must stop accepting input when the user enters a four-letter word. Assume that no word is more than 20 letters long. An interactive session with the program might look like this:

```
Enter word: dog
Enter word: zebra
Enter word: rabbit
Enter word: catfish
Enter word: walrus
Enter word: cat
Enter word: fish

Smallest word: cat
Largest word: zebra
```

Hint: Use two strings named *smallest_word* and *largest_word* to keep track of the "smallest" and "largest" words entered so far. Each time the user enters a new word, use *strcmp* to compare it

with *smallest_word*; if the new word is "smaller", use *strcpy* to save it in *smallest_word*. Do a similar comparison with *largest_word*. Use *strlen* to determine when the user has entered a four-letter word.

Questions:

1) Attach the source code of your C program into the answer sheet.

```
#include <stdio.h>
#include <string.h>

int main(){
	char input[21], smallest_word[21],largest_word[21];
	printf("BEWARE! THIS PROGRAM ENDS ONCE A 4-LETTER WORD IS INPUTTED BY THE USER! \n");
	printf("Enter word: ");
	scanf("%s",input);

	strcpy(smallest_word,input);
	strcpy(largest_word,input);

	while(strlen(input) != 4){
		if(strcmp(input,smallest_word) < 0){
			strcpy(smallest_word,input);
		}
		else if(strcmp(input,largest_word) > 0) {
			strcpy(largest_word,input);
		}

		printf("Enter word: ");
		scanf("%s",input);
	}

	printf("Smallest Word: %s\n", smallest_word);
	printf("Largest Word: %s\n", largest_word);

	return 0;
}
```

2) Run the C program, attach a screenshot of the output in the answer sheet. 1

```
[mpatel185@gsuad.gsu.edu@snowball ~]$ vi findStr.c
[mpatel185@gsuad.gsu.edu@snowball ~]$ [mpatel185@gsuad.gsu.edu@snowball ~]$ gcc -o findStr findStr.c
[mpatel185@gsuad.gsu.edu@snowball ~]$ ./findStr
BEWARE! THIS PROGRAM ENDS ONCE A 4-LETTER WORD IS INPUTTED BY THE USER!
Enter word: dog
Enter word: zebra
Enter word: rabbit
Enter word: catfish
Enter word: walrus
Enter word: cat
Enter word: fish
Smallest Word: cat
Largest Word: zebra
[mpatel185@gsuad.gsu.edu@snowball ~]$
```

Submssion:

- Please follow the instructions below step by step, and then write a report by answering the questions and upload the report (named as Lab10_FirstNameLastName.pdf or Lab10_FirstNameLastName.doc) to Google Classroom, under the rubric Lab 10 – Post Lab Assignment.
- Upload the C files findStr.c to the folder named "Lab 10 – Post Lab" in Google Classroom.
- Please add the lab assignment NUMBER and your NAME at the top of your filesheet.